اصول پردازش تصویر

نیم‌سال اول ۱۳۹۹-۱۴۰۰

مدرس: دکتر مصطفی کمالی تبریزی

دانشکده علوم ریاضی

تمرین سری دوم ـ سوال چهارم

نام و نام‌خانوادگی: سیدعلیرضا خادم

شماره دانشجویی: ۹۷۱۰۰۳۹۸

موارد لازم.

برای اجرا لازم است تا تصاویر far.jpg و near.jpg در مسیر /EX2_Q4/images قرار داشته باشد.

روند کلی حل.

در ابتدا دو تصویر را با استفاده از تابع match از فایل match.py هم اندازه کرده و با هم منطبق نمایید، یعنی قسمت های مشابه یا معادل را روی هم قرار دهید. برای مثال، اگر بخواهید دو تصویر از صورت دو شخص ر ا باهم ادغام کنید، بهتر است در ابتدا دو تصویر را طوری تغییر دهید تا اجزای متناظر صورت دو شخص در یک مکان از تصویرشان قراردادیم. به این صورت که محور چرخ‌های جلو و عقب و یک نقطه از باک دو موتور را در نظر گرفتیم و با استفاده از Affin Tranformation تصویر دو موتور را بر هم منطبق کردیم.

جزئیات یک تصویر در حوزه فرکانس با فرکانس های بالا مشخص می شوند. بنابراین، در تصویری که می خواهیم از نزدیک دیده شود باید فرکانس های پایین را حذف کرده و فرکانس های بالا را نگه داریم. وقتی از دور به یک تصویر نگاه می کنیم، بیشتر کلیات تصویر که با فرکانس های پایین مشخص می شوند دیده می شود. بنابراین، در تصویری که می خواهیم از دور دیده شود فرکانس های بالا را حذف نموده و فرکانس های پایین را نگه داریم. برای تر کیب دو تصویر، ابتدا هر دو تصویر را به دامنه فرکانس میبریم. سپس، در تصویر او ل (تصویری که می خواهیم از نزدیک دیده شود) فرکانس های پایین را حذف نموده و فرکانس های بالا را نگه می‌داریم. در تصویر دوم (تصویری که می خواهیم از دور دیده شود) فرکانس های بالا را حذف نموده و فرکانس های پایین را نگه می‌داریم. این دو تصویر در دامنه فرکانس را با هم جمع می‌کنیم. در نهایت، تصویر هیبریدی حاصل را به حوزه مکان برمی‌گردانیم.

توضیح کد.

برنامه در مجموع حاوی ۳ فایل با فرمت py. می‌باشد که توضیحات هر فایل در پایین آمده است.

- **utilities.py**

gaussian_mask(sigma, height, width, mu=0.0)

این تابع sigma ، height ، width و mu را به عنوان ورودی می‌گیرد و یه فیلتر گوسی با انحراف معیار sigma ، میانگین mu (که به طور پیش‌فرض ۰ است) و با ابعاد داده شده برمی‌گرداند.

save_fft_channel_images(..., ..., channel_name)

این تابع near_image_shifted_fft که عکسِ نزدیک که به حوزه فرکانس رفته و شیفت داده شده است و far_image_shifted_fft که عکسِ دور است که به حوزه فرکانس رفته و شیفت داده شده است را به عنوان ورودی می‌گیرد و بزرگی، لگاریتم بزرگی و اسکیل شده لگاریتم بزرگی به بازه ۰ تا ۲۵۵ را محاسبه می‌کند و با توجه به ورودی channel_name که مشخص می‌کند این داده‌ها مربوط به کدام کانال است، لگاریتم بزرگیِ اسکیل شده را در مسیر /EX2_Q4/images با فرمت‌ای که مشاهده می‌کنید ذخیره می‌کند.

save_filters(near_filter, far_filter, r, s)

این تابع فیلترهایی که برای تصاویر نزدیک و دور در نظر گرفته شده است را با sigma ی مربوط به آن‌ها را به عنوان ورودی می‌گیرد و فیلترها را اسکیل می‌کند تا قابل مشاهده باشد و با فرمتی که خواسته شده در مسیر /EX2_Q4/results ذخیره می‌شوند.

apply_circle_mask(src_image, radius, inverse)

این تابع تصویر src_image را به عنوان ورودی می‌گیرد و یک دیسک با شعاع radius و به مرکز تصویر در نظر می‌گیرد و با توجه به اینکه مقدار inverse ، True یا False است یک آرایه با ابعاد تصویر ورودی که به ترتیب داخل دیسک یا خارج دیسک همان مقادیر تصویرِ ورودی و خارجِ ناحیه مدنظر همه درایه‌ها صفر باشد را به عنوان خروجی برمی‌گرداند.

save_filters_cutoff(near_filter, far_filter, near_radius, far_radius)

این تابع فیلتر مربوط به تصویر نزدیک و فیلتر مربط به تصویر دور را با دو شعاع که به ترتیب متناظر با این فیلترها هستند را ورودی می‌گیرد. در فیلتر بالاگذر، ضرایبی که فاصله آن ها تا مبدأ بیشتر از این مقدار near_radius می باشند حفظ شده و بقیه ضرایب مساوی صفر قرار داده می شوند. در فیلتر پایین گذر، ضرایبی که فاصله آن ها تا مبدأ کمتر از این مقدار far_radius باشند حفظ شده و بقیه ضرایب مساوی صفر قرار داده می شوند.هر دو تصویر در دامنه فرکانس در یک نوار غیر صفر می شوند. برای ترکیب دو تصویر در این نوار، از میانگین گیری استفاده شده‌است.این تابع فیلترهای حاصل را با نام های Q4_09_highpass_cutoff.jpg و Q4_10_lowpass_cutoff.jpg به ترتیب برای فیلترهای بالاگذر و پایین گذردر مسیر /EX2_Q4/results ذخیره می‌کند.

create_hybrid(far_image, near_image, channel_name)

این تابع تصویر نزدیک و تصویر دوری که به عنوان ورودی گرفته است را در ابتدا با استاده از تابع np.fft.fft2 به حوزه فرکانس می‌برد بعد با استفاده از تابع np.fft.fftshift آن را در حوزه فرکانس به گونه‌ای شیفت می‌دهد تا کمترین فرکانس در مرکز تصویر قرار بگیرد. در ادامه فیلتر بالاگذر را به صورت gaussian_mask(0.04, rows, cols) - 1 و فیلتر پایین‌گذر را به صورت gaussian_mask(0.07, rows, cols) نظر می‌گیریم. با استفاده از تابع save_filters_cutoff ، برای فیلتر بالاگذر شعاع ۱۰ و برای فیلتر پایین‌گذر شعاع ۳۵ را اعمال می‌کنیم. در مرحله بعد هر یک از تصاویر که در حوزه فرکانس شیفت داده شده‌اند را در فیلتر مربوط به خودشان مؤلفه به مؤلفه ضرب می‌کنیم. ریزالت در حوزه فرکانس را به صورت جمع fil_near_image_fft و fil_far_image_fft می‌نویسیم. در نهایت هم معکوس شیفت و معکوس fft را روی ریزالت اعمال می‌کنیم تا به حوزه مکان برگردیم.

بقیه توابع این فایل برای ذخیره‌سازی نتایج با فرمتی که در سوال آماده پیاده سازی شده و نکته خاصی ندارند.

match.py ○

match

این تابع دو تصویر را به عنوان ورودی می‌گیرد و بااستفاده از دو مجموعه نقاطی که نشان دهنده نقاط متناظر تو تصویر هستند یک Affine Transformation بین تو تصویر پیدا میکند و تصویر اول را روی تصویر دوم منطبق می‌کند.

q۴.py ○

در این فایل ابتدا تصاویر نزدیک و دور لود شده‌اند و بعد با استفاده از تابع match تصویر نزدیک بر تصویر دور منطبق شده است. در ادامه با استفاده از تابع cv.split کانال‌های رنگی مربوط به هر تصویر تفکیک شده و به صورت مجزا برای هر کانال رنگی با استفاده از تابع create_hybrid تصویر هیبریدی ساخته شده؛ در نهایت هم این کانال‌های هیبرید شده با استفاده از تابع cv.merge با هم مرج شده‌اند تا تصویر نهایی ایجاد شود.

در این میان یه سری تصویر با فرمتی که در صورت سوال آمده است ایحاد شده و در مسیر EX2_Q4/results/ ذخیره شده‌اند.